

Reducing Uncertainties in Neural Network Jacobians and Improving Accuracy of Neural Network Emulations with NN Ensemble Approaches

Vladimir Krasnopolsky

Abstract—A new application of the NN ensemble technique to improve the accuracy and stability of the calculation of NN emulation Jacobians is presented. The term “emulation” is defined to distinguish NN emulations from other NN models. It was shown that, for NN emulations, the introduced ensemble technique can be successfully applied to significantly reduce uncertainties in NN emulation Jacobians to reach the accuracy sufficient for the use in data assimilation systems. An NN ensemble approach is also applied to improve the accuracy of NN emulations themselves. Two ensembles linear, conservative and nonlinear (uses an additional averaging NN to calculate the ensemble average) were introduced and compared. The ensemble approaches: (a) significantly reduce the systematic and random error in NN emulation Jacobian, (b) significantly reduces the magnitudes of the extreme outliers and, (c) in general, significantly reduces the number of larger errors, (d) nonlinear ensemble is able to account for nonlinear correlations between ensemble members and improves significantly the accuracy of the NN emulation as compared with the linear conservative ensemble in terms of systematic (bias), random, and larger errors.

I. INTRODUCTION

The simplest multi-layer perceptron (MLP) neural network (NN) is a generic analytical nonlinear approximation or model for nonlinear (continuous) mappings [1]. The MLP NN uses for the approximation of mappings a family of functions like:

$$y_q = a_{q0} + \sum_{j=1}^k a_{qj} \cdot \tanh(b_{j0} + \sum_{i=1}^n b_{ji} \cdot x_i); \quad q = 1, 2, \dots, m \quad (1)$$

where x_i and y_q are components of the input and output vectors respectively, a and b are fitting parameters.

A mapping between two vectors X (input vector) and Y (output vector) or target mapping can be symbolically written as:

$$Y = M(X); \quad X \in \mathfrak{R}^n, Y \in \mathfrak{R}^m \quad (2)$$

A large number of important practical geosciences applications may be considered mathematically as a mapping (2) [2-9]. The regular NN approximation technique provides a NN approximation with sufficiently small approximation errors on the training set; however, it does

not guarantee, without additional constraints, a satisfactory generalization (interpolation) capability of the NN approximations. However, in some applications [2-8] the necessity arises to improve the generalization (interpolation) accuracy of developed NN approximations. We have developed a NN emulation technique to emulate these mappings with high approximation and interpolation accuracy [2-6] (see also section 2 of this paper). In some applications, developed NN emulations of mappings (2) are used for inversion [3], in a data assimilation system (DAS) [7] or for error or/and sensitivity analysis. In all these cases not only the mapping (2) but also its first derivative are used. It means that the NN emulation Jacobian that is a matrix of the first derivatives of the outputs of the NN emulation (1)

over its inputs, $\left\{ \frac{\partial y_q}{\partial x_i} \right\}_{i=1, \dots, n}^{q=1, \dots, m}$, has to be calculated. From a

technical point of view, the calculation of the Jacobian is almost trivial. It is performed by an analytical differentiation of eq. (1). However, from a theoretical point of view, the inference of the NN Jacobian is an ill-posed problem [10] which leads to significant uncertainties in calculated NN Jacobians [11-13].

For such applications that require an explicit calculation of the NN Jacobian, several solutions have been offered and investigated to reduce the NN Jacobian uncertainties: (i) the Jacobian can be trained as a separate additional NN [4]; (ii) the mean over the data set Jacobian can be calculated and used [11]; (iv) regularization techniques like “weight smoothing” [12] or the technique based on a principle component decomposition [13] can be used to stabilize the Jacobians; (v) the Jacobian can be trained: included in the training data set and as actual additional outputs in the NN; the error (or cost) function, which is minimized in the process of NN training, can be modified to accommodate the Jacobian; in other words, the Euclidian norm, which is usually used for calculating the error function, should be changed to the first order Sobolev’s norm. Actually, *Hornik et al.* [14] showed that the function of the Sobolev’s space can be approximated with all their derivatives. This and other similar theoretical results are very important because they prove the existence of the approximation; however, they do not suggest explicit approaches.

In this paper we introduce a new, NN ensemble approach to reduce uncertainties in calculated NN emulation Jacobians. NN ensemble approaches have been introduced

Vladimir Krasnopolsky is with the National Centers for Environmental Prediction, Camp Springs, MD 20746 USA; (corresponding author V. Krasnopolsky phone: 301-763-8000 x 7262; fax: 301-763-8545; e-mail: Vladimir.Krasnopolsky@noaa.gov).

by many authors [15-20]. They were used to improve NN classification, NN approximation and NN generalization abilities. To the best of our knowledge, this work is the first one that introduces an application of a NN ensemble technique for reducing uncertainties of NN emulation Jacobians.

In this paper we also compare a conservative ensemble [20] where simple (with equal weights for all members) averaging of the members provides the ensemble mean and other statistics with a nonlinear ensemble where an averaging NN is introduced that takes into account nonlinear correlations between ensemble members. This averaging NN, given ensemble members as inputs, generates a nonlinear ensemble average.

In section 2 of this paper we define NN emulation of complex mappings; we also discuss a generic approach to use NN ensembles to improve the accuracy of the NN emulation Jacobian and the NN emulation. In section 3 we illustrate these approaches using as a test bed a particular practical application that is described in details in [7], a NN emulation for the ocean surface elevation mapping in an ocean numerical model. Conclusions are presented in section 4.

II. BACKGROUND: NN EMULATIONS AND NN ENSEMBLE APPROACHES TO NN EMULATIONS AND NN EMULATION JACOBIANS

A. NN Emulations of Complex Mappings

In this paper, we use the terms an **emulating NN** or a **NN emulation** for NN (1) that provides a functional emulation of the target mapping (2) that implies a small approximation error for the training set and smooth and accurate interpolation between training set data points inside the mapping domain D . The term “emulation” is introduced to distinguish between these NNs and approximating NNs or NN approximations that guarantee small approximation error for the training set only.

When an emulating NN is developed, in addition to the criterion of small approximation error at least three other criteria are used: (i) the NN complexity (proportional to the number k of hidden neurons when other topological parameters are fixed) is controlled and restricted to a minimal level sufficient for good approximation and interpolation; (ii) independent validation and test data sets are used in the process of training (validation set) to control overfitting and after the training (test set) to evaluate interpolation accuracy; (iii) redundant training set (additional redundant data points are added in-between training data points sufficient for a good approximation) is used for improving the NN interpolation abilities.

The correspondence between the emulating NN complexity and the target mapping complexity is usually better than for an approximating NN with the same approximation error. An emulating NN complexity is usually close to the minimal one; thus, the emulating NN is

usually faster. It usually provides better interpolation generalization and better resolution of the target mapping at the same approximation accuracy.

B. Multiple NN Emulation Solutions and Ensemble Approach

As a nonlinear model or nonlinear approximation of the mapping (2), the NN approximation problem allows for multiple solutions or for multiple NN emulations (1) for the same mapping (2). Existence of multiple solutions is a common property of nonlinear models, or of nonlinear approximations. These models have nonlinear parameters that may be changed to generate solutions, which may be close in terms of satisfying to a particular criterion (e.g., approximation error) used for obtaining the solutions. For example, the same mapping (2) can be approximated with NNs (1) with different numbers of hidden neurons, with different weights (resulting from the NN training with different initializations), different partitions of the training set, etc. At the same time, these multiple models (NNs) may be different in terms of other criteria providing complementary information about the target mapping (2). The availability of multiple solutions may lead to some inconveniences like a necessity to introduce an additional step that is to use additional criteria to select a single, optimal model or problems like uncertainties or multiple solutions for the NN emulation Jacobian. The existence of multiple and significantly different solutions for the NN Jacobian is a consequence of the fact that the statistical inference of the NN Jacobian is an ill-posed problem [10].

On the positive side, availability of multiple models (NN emulations and NN emulation Jacobians), providing complimentary information about the target mapping (2) and its Jacobian, opens an opportunity to use ensemble approaches. It allows for integrating the complimentary information, containing in the ensemble members, into an ensemble that “knows” more about or represents the mapping (2) and its Jacobian better than each of the individual ensemble members (a particular NN emulation, or a particular NN emulation Jacobian).

An ensemble of NNs consists of a set of members, that are individually trained NNs. They are combined when applied to a new input data to improve the generalization (interpolation) ability. The previous research showed that an ensemble is often more accurate than any or most of the individual member of the ensemble.

The previous research also suggests that any mechanism that causes some randomness in or perturbation for the formation of NN ensemble members can be used to form an accurate NN ensemble [16]. For example, ensemble members can be created by training different members: (a) on different subsets of the training set [16]; (b) on different subdomains of the training domain; (c) using NNs with different topology (different number of hidden neurons) [17]; (d) using NNs with the same architecture but with different initial conditions for NN weights [18,19].

In the context of our application, i. e. approximation of a complex mapping (2), the members of the ensemble are separately trained NNs which provide emulations for the target mapping with slightly different approximation and interpolation accuracies. Because of the properties of the NN emulations described in subsection A, we can expect that these emulations do not oscillate strongly between the training set data points. It means that the spread of the emulation accuracy and, what is even more important, the spread between different solutions for the NN Jacobian (the Jacobian uncertainties) are limited. We can expect that the ensemble average will provide a better approximation and interpolation than its individual members.

C. Conservative Ensemble vs. Nonlinear NN ensemble

Different ways of combining NN ensemble members into the ensemble have been developed and investigated [15]. In this work, we start from using a conservative ensemble [20] where simple (with equal weights for all members) averaging of the members provides the ensemble mean and other statistics. The conservative ensemble or its modifications (linear averaging with non-equal weights) are mostly popular in applications. However, these approaches cannot account for possible nonlinear correlations between ensemble members. In this paper we also use a nonlinear averaging approach, a NN ensemble averaging, with an averaging NN that takes into account nonlinear correlations between ensemble members. This NN is trained given ensemble members as inputs to generate a nonlinear ensemble average. It is shown that this approach may additionally (as compared with the conservative ensemble) significantly reduce the random and systematic components of approximation and interpolation errors.

III. APPLICATIONS OF THE NN ENSEMBLE APPROACH

Here we introduce an application of the NN ensemble approach in the context of our NN application described in [7] – the NN emulations for functional nonlinear dependencies and mappings between atmospheric or ocean state variables that are implicitly contained in the highly nonlinear coupled partial differential equations of an atmospheric or ocean dynamical model and, therefore, in numerical outputs of these models.

In particular, in a layered ocean model the sea surface height (SSH or η) signal depends in part on the disposition of the layers in a vertical column. Therefore, this dependence, after emulation it with NN, can be written as

$$\eta_{NN} = \phi_{NN}(X) \quad (3)$$

where ϕ_{NN} is a NN and X is a vector that represents a complete set of state variables, which determines SSH. In this particular case the vector X was selected as

$X = \{I, \theta, z_{mix}\}$, where I is the vector of interfaces (vertical coordinates used in ocean model HYCOM, see [7] for description of the model), θ is the vector of potential temperature, and z_{mix} is the depth of the ocean mixed layer (a total of 50 variables).

An analytical NN emulation (3) for the relationship between model state variables, X , and sea surface height, η , was derived using the simulated model fields which are treated as error free data [7]. A simulation that covers almost two years (from Julian day 303, 2002 to 291, 2004) was used to create training, validation and test data sets. The periods covered by these data sets and their sizes are shown in Table 1. Each data set consists of records $\{\eta_i, X_i\}_{i=1, \dots, N}$ collocated in space and time and uniformly distributed over the model domain.

TABLE 1. PERIODS COVERED BY TRAINING, VALIDATION AND TEST DATA SETS AND THEIR SIZES.

Set	Beginning Date (Julian day, year)	End Date (Julian day, year)	Size, N (number of profiles)
Training	303, 2002	52, 2004	563,259
Validation	303, 2002	52, 2004	563,259
Test	53, 2004	291, 2004	563,259

In the context of the problem described in the previous subsection, the NN ensemble approach leads to the following solution. The complexity of the NN emulation (3) was limited; only three hidden neurons were allowed. Then ten NN emulations (3) with the same number of the hidden neurons (three) were trained using differently perturbed initial conditions for the NN weights. As a result, a NN ensemble that consists of ten members, ten NN emulations with identical architecture (50 inputs, 3 hidden layers, and 1 output) but different weights and different approximation accuracies, has been created.

A. NN Ensembles for Reducing Uncertainties of the NN Jacobian

The NN emulation (3) can be used in the ocean DAS to enhance assimilating SSH and to improve the propagation of the surface SSH signal to other vertical levels and other variables. In the ocean DAS the increment of the SSH, $\Delta\eta$,

$$\Delta\eta_{NN} = \sum_{i=1}^n \left. \frac{\partial \phi_{NN}}{\partial X_i} \right|_{X=X_0} \cdot \Delta X_i \quad (4)$$

where ΔX_i are increments of state variables, X_0 is an initial value of state variables and n is the dimensionality of the

vector X (the number of inputs of the NN emulation (3)). Then the calculated $\Delta\eta_{NN}$ is compared with the observed $\Delta\eta_{obs}$ and the difference is used to adjust ΔX .

The quality of the single NN Jacobian may not be sufficient for the use in DAS applications. However, an ensemble approach can be used to improve the NN Jacobian calculations. The NN ensemble described above was used

$$\left\{ \frac{\partial \phi_{NN}^j}{\partial X_i} \right\}_{j=1, \dots, ens}$$

here to create an ensemble of ten NN Jacobians where $ens = 10$ is the number of the ensemble members. Then the ensemble average Jacobian has been calculated,

$$\overline{\frac{\partial \phi_{NN}}{\partial X_i}} = \frac{1}{ens} \sum_{j=1}^{ens} \frac{\partial \phi_{NN}^j}{\partial X_i}, \quad i = 1, \dots, n \quad (5)$$

Now, eq. (4) was used to calculate $\Delta\eta_{NN}$ using each ensemble member Jacobian and the ensemble average Jacobian (5). These values of $\Delta\eta_{NN}$ were compared with exact $\Delta\eta$ known from the model simulation.

This comparison technique was applied to the last day of the entire model simulation. This day is separated by the time interval of about 8 month from the last day of simulation used for NNs training and validation. Fields generated by the model at 00Z were used to create inputs, X , for the NN emulation Jacobians. Then the NN emulation Jacobian ensemble members were applied (4) over the entire domain (with coastal areas excluded) to generate an ensemble of 2-D field of $\Delta\eta_{NN}^j$ and calculate $\Delta\eta_{NN}$ using in (4) the ensemble average Jacobian (5). Also a non-dimensional distance in the model state space between vectors X_0 and $X = X_0 + \Delta X$ was introduced,

$$S = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{\Delta X_i}{X_0^i} \right)^2} \quad (6)$$

These fields were compared with the corresponding field of SSH, η , generated by the model. We performed also multiple case studies for particular locations inside the model domain. Results of one of this study are presented in Figs. 1-4.

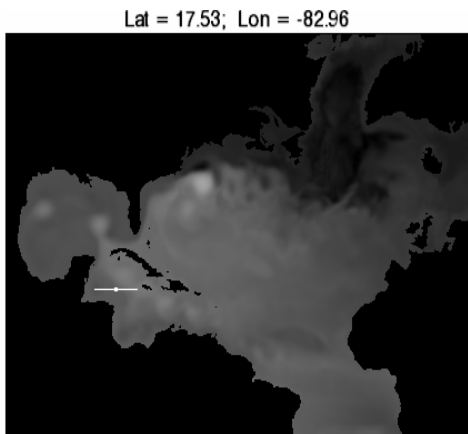


Fig.1. The location of the cross section (white horizontal line) inside the model domain; white dot show the position of X_0 .

Fig. 1 shows the location of the cross section (white horizontal line) inside the model domain; white dot show the position of X_0 . Starting from this position we moved left and right grid point by grid point using X values at these grid points to calculate ΔX and the non-dimensional distance in the model state space, S (6). These values of ΔX were used in (4) to calculate $\Delta\eta$.

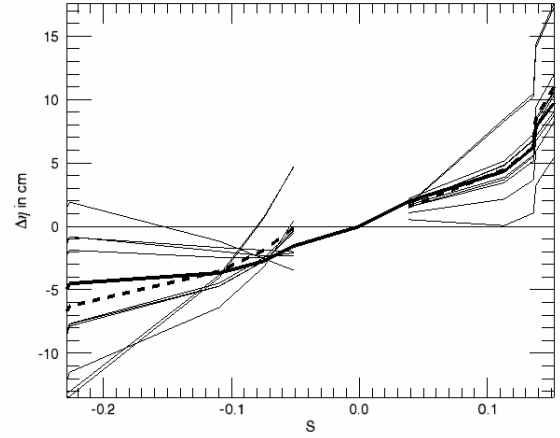


Fig.2. $\Delta\eta$ calculated using (4) and the NN ensemble member Jacobians (an envelope of thin solid lines that illustrates the Jacobian uncertainties), exact $\Delta\eta$ calculated from the model (thick solid line), and $\Delta\eta$ calculated using the ensemble average Jacobian (5) (thick dashed line). $\Delta\eta$ is shown vs. the distance in the model state space, S (6).

Fig. 2 shows $\Delta\eta$ calculated using (4) and the NN ensemble member Jacobians (an envelope of thin solid lines that illustrates the Jacobian uncertainties), exact $\Delta\eta$ calculated from the model (thick solid line), and $\Delta\eta$ calculated using the ensemble average Jacobian (5) (thick dashed line). $\Delta\eta$ is shown vs. the distance in the model state space, S . This figure demonstrates how significantly using of the ensemble average improves the NN Jacobian. The larger are the distances S the more significant is the reduction of the Jacobian uncertainties.

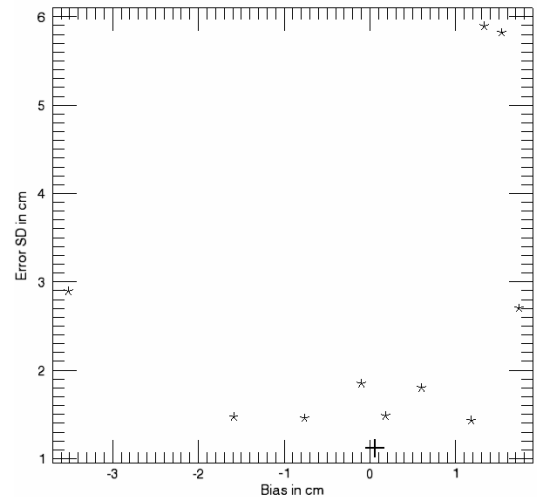


Fig.3. The systematic error (bias) and the random error (error standard deviation) for $\Delta\eta$ calculated along the path shown in the upper right panel using (4). The asterisks correspond to errors when the ensemble member Jacobians were used in (4), the cross corresponds to the case when the ensemble average Jacobian (5) was used.

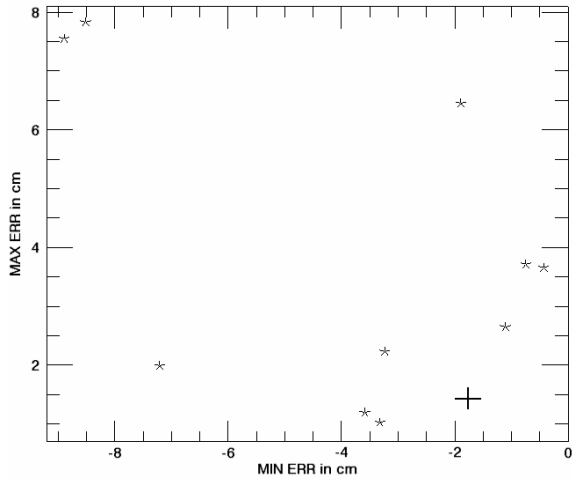


Fig.4. The minimum and maximum errors along the path. The asterisks correspond to errors when the ensemble member Jacobians were used in (4), the cross corresponds to the case when the ensemble average Jacobian (5) was used.

Fig. 3 shows the systematic error (bias) and the random error (error standard deviation) for $\Delta\eta$ calculated along the path shown in the upper right panel using (4). The asterisks correspond to errors when the ensemble member Jacobians were used in (4), the cross corresponds to the case when the ensemble average Jacobian (5) was used. The ensemble bias is equal to the mean bias of the members as it can be expected using this simple method of calculating the ensemble average. This figure also shows that in the case of Jacobian the ensemble approach is very effective in reducing random errors; it shows that ensemble random error (1.1 cm) is less than the random error of any of the ensemble members. The reduction in systematic ($\sim 90\%$) and random ($\sim 65\%$) errors with respect to the maximum single member errors is very significant.

Fig. 4 shows minimum and maximum errors along the path or statistics for extreme outliers. When each ensemble member NN Jacobian is applied in (4), for each particular input the NN produces an error. Among all these errors there exist one largest negative (or minimum) error and one largest positive (or maximum) errors or two extreme outliers that demonstrate the worst case behavior (scenario) that we can expect from this particular NN emulation. These two extreme outliers (negative and positive) are presented as a star for each NN member in the figure. The ensemble average Jacobian (5), when used in (4), also generates such two extreme outliers that are presented as the cross in the figure. The figure shows that the NN ensemble approach is also an effective tool in reducing (~ 4 times) large errors in NN Jacobians.

Next we applied the same procedure at all grid points of the model domain. The errors have been calculated along numerous paths (horizontal and vertical) all over the model domain. Fig. 5 shows RMS error in $\Delta\eta$ as a function (binned and averaged in each bin) of non-dimensional distance S over entire domain. Thin lines correspond to the ensemble members (an envelope of thin solid lines illustrates the Jacobian uncertainties) and the thick line shows the ensemble result. The ensemble significantly improves

statistics at all considered distances S . The ensemble is always better than the best ensemble member.

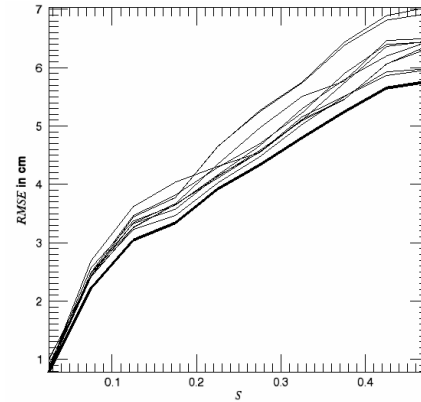


Fig.5. Errors in $\Delta\eta$ as a function (binned and averaged in each bin) of non-dimensional distance S over entire model domain. Thin lines correspond to the ensemble members and the thick line shows the ensemble result.

To better understand the magnitudes of errors presented in this and the next section, these magnitudes should be compared with the errors in observed satellite data $\Delta\eta_{obs}$ assimilated in the oceanic DAS [7]. The accuracy of the observed data is about 5 cm. It means that our NN emulation (3) and ensemble techniques allow us to reduce the Jacobian uncertainties and produce the ensemble Jacobian (5) that is sufficiently accurate to be used in ocean DASs.

B. Ensemble Approach to Improve Emulation Accuracy; Linear and Nonlinear Ensembles

Here we apply the NN ensemble in a more traditional mode to improve the accuracy of the NN emulation (3) (see also [8]). After the NN ensemble (see above) was created, each NN member (that is a particular realization of the NN emulation (3)) was applied to the test set and the error statistics for each NN member was calculated and plotted in Fig. 6. The vertical axis of the figure shows the random part of the approximation error (the standard deviation of the error) and the horizontal axis – the absolute value of the systematic error (bias). Both errors are normalized to the corresponding maximum member error (maximum member bias or maximum member error standard deviation). Each ensemble member is presented by a star at this figure. The figure illustrates the spread of the ensemble members; it is significant. For different members, the systematic error changes about 25% and the random error – about 10%.

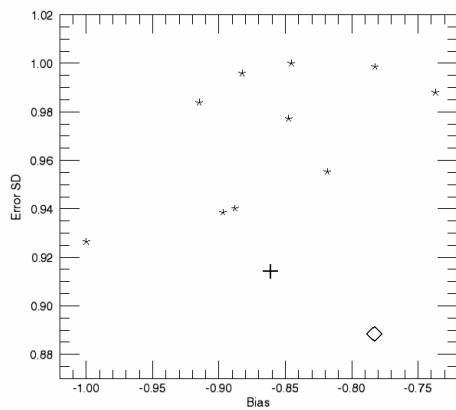


Fig. 6. Vertical axis shows the random part of the emulation error (the standard deviation of the error) normalized to the maximum member error and the horizontal axis – the systematic error (bias) also normalized to the maximum member error. Each ensemble member is represented by a star, the conservative ensemble average – by the cross, and the nonlinear ensemble using the averaging NN by diamond at the figure.

The next step was to produce the ensemble average. Ensemble average can be produced in different ways [20]. The first averaging approach that we used here is the simplest and linear method of ensemble averaging – conservative ensemble [20]. Each of ten members of the NN ensemble was applied to the test set record by record. Thus, for each record, for each set of inputs, ten NN outputs were produced. Then, the mean value (in a regular statistical sense) of these ten numbers was calculated and used to compare with the exact output to calculate ensemble statistics presented by the cross in Fig. 6. The ensemble bias is equal to the mean bias of the members as it can be expected using this simple linear method of calculating the ensemble average. Fig. 6 also illustrates a known fact that ensemble approaches are very effective in reducing random errors; it shows that ensemble random error is less than the random error of any of the ensemble members. The reduction in systematic (~15%) and random (~9%) errors with respect to the maximum single member errors is not large but significant.

Conservative ensemble is simple; however, it is linear; it completely neglects nonlinear correlations and dependencies between ensemble members. To estimate the contribution of these nonlinear correlations and to use these contributions to improve ensemble averaging we developed a nonlinear ensemble that uses an additional averaging NN to calculate the ensemble average. Schematically this approach is illustrated in Fig. 7. The inputs of the averaging NN are constituted from the outputs of the ensemble member NNs. The number of inputs of the averaging NN is equal to the number of ensembles multiplied by the number of outputs in a single ensemble member NN (10 in our case). It has the same outputs as a single ensemble member NN (one in our particular case). The averaging NN was trained using training and

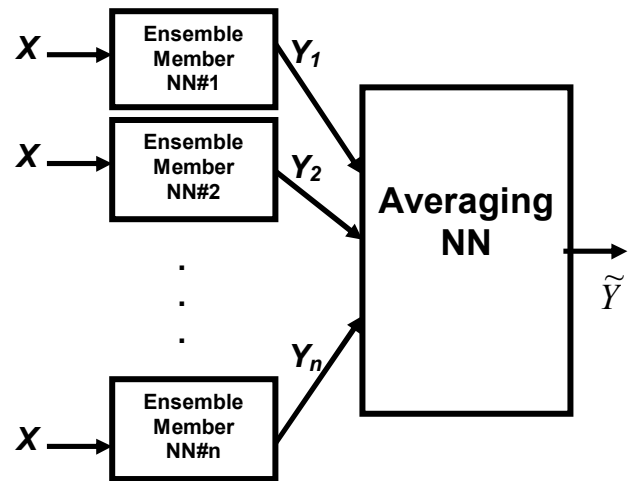


Fig. 7 Schematic representation of a nonlinear ensemble that uses an averaging NN. A tilde over the averaging NN output \tilde{Y} emphasizes that a nonlinear ensemble average is produced. X is an input of the emulation (3).

validation sets prepared on the basis of the training and validation sets used for training the ensemble member NNs. The test statistics presented here were calculated using the test set.

Fig. 6 shows statistics for nonlinear ensemble using the averaging NN with diamond. It shows that the magnitude of the nonlinear correlations between ensemble members is significant and can be successfully used to improve ensemble accuracy. Comparison of the position of the cross and the diamond in Fig. 6 shows that, as compared with the conservative ensemble, the nonlinear ensemble gives an additional improvement in bias of order of 10%. The nonlinear ensemble bias is close to the minimum ensemble member bias. An additional improvement in the random error is a bit smaller (about 5%) but significant.

Fig. 8 shows statistics for extreme outliers. When each ensemble member NN is applied to the test set, for each record the NN produces an error. Among all these error there exist one largest negative (or minimum) error and one largest positive (or maximum) errors or two extreme outliers that demonstrate the worst case behavior (scenarios) we can expect from this particular NN emulation. These two extreme outliers are presented as a star for each NN member in Fig. 8. Each ensemble also generates such two extreme outliers that are presented as the cross for the conservative ensemble and a diamond for the nonlinear ensemble in Fig. 8.

Fig. 8 shows that the NN ensemble approach is an effective tool in reducing extreme outliers (~25%). However, a careful analysis of the figure reveals also very interesting features of the statistics presented in this figure. The distribution of starts shows a significant spread. It also demonstrates a significant clustering and correlation between extreme outliers produced by ensemble members. These facts and the position of the conservative ensemble (cross) in the figure suggest that the members of the ensemble are nonlinearly correlated. A significant improvement introduced by the nonlinear ensemble (diamond) supports this conclusion.

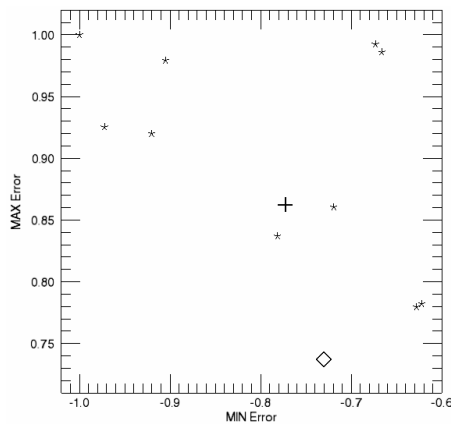


Fig.8. Extreme outliers statistics. Vertical axis shows the largest positive (or maximum) and the horizontal axis – the largest negative (or minimum) emulation error over the entire test set. Each ensemble member is represented by a star, the conservative ensemble – by the cross, and the nonlinear ensemble – by the diamond at the figure.

IV. CONCLUSION

In this paper, we have presented a new application of the NN ensemble technique to reduce the uncertainties and improve the accuracy of the NN emulation Jacobian. We defined the term “emulation” and show why it is important to distinguish NN emulations from other NN models. We introduced an ensemble technique and showed that, for NN emulations, this ensemble technique can be successfully applied to significantly reduce uncertainties in NN emulation Jacobians. In the framework of an ocean data assimilation application [7] we showed that our ensemble approach allows to calculate the NN emulation Jacobian with the accuracy sufficient for the use in the data assimilation system. The ensemble approach: (a) significantly reduces the systematic and random error in NN emulation Jacobian, (b) significantly reduces the magnitudes of the extreme outliers and, (c) in general, significantly reduces the number of larger errors.

Here and in [8] we have also applied the NN ensemble approach to improve the emulation accuracy of NN emulations of complex multidimensional mappings. In particular, in [8] we applied this technique to NN emulations we developed for the LWR parameterization of NCAR CAM. In this paper, we applied the NN ensemble technique to a mapping developed in the framework of an ocean data assimilation application [7]. This mapping and the corresponding NN emulation relate the sea surface elevation to a vector of oceanic state variables. We introduced and compared two NN ensembles: (1) a linear conservative ensemble estimating the ensemble average as a simple linear mean of the ensemble members, and (2) nonlinear NN ensemble that uses a special NN to estimate a nonlinear ensemble average given ensemble members. We have shown that practically all individual NN emulations that we have trained in the process of development of an optimal NN emulation, can be used, within the NN ensemble approach

for improving generalization (interpolation) ability of our NN emulations: (a) significantly reducing the systematic and random interpolation error, (b) significantly reducing the magnitudes of the extreme outliers and, (c) in general, significantly reducing the number of larger errors. It was also shown that nonlinear ensemble is able to account for nonlinear correlations between ensemble members and improves significantly the accuracy of the NN emulation as compared with the linear conservative ensemble in terms of systematic (bias), random, and larger errors.

ACKNOWLEDGMENT

The author thanks Dr. M. S. Fox-Rabinovitz for stimulating discussions and Dr. C. Lozano for discussions of an oceanic application used in this work.

REFERENCES

- [1] K. Funahashi, “On the Approximate Realization of Continuous Mappings by Neural Networks”. *Neural Networks*, 2, pp. 183-192, 1989.
- [2] V. Krasnopolsky, L. C. Breaker, and W. H. Gemmill, “A neural network as a nonlinear transfer function model for retrieving surface wind speeds from the special sensor microwave imager”, *J. Geophys. Res.*, 100, pp. 11033–11045, 1995.
- [3] V. M. Krasnopolsky, W. H. Gemmill, and L. C. Breaker, “A multi-parameter empirical ocean algorithm for SSM/I retrievals”, *Canadian Journal of Remote Sensing*, 25, pp. 486–503, 1999.
- [4] V.M. Krasnopolsky, D.V. Chalikov, and H.L. Tolman, “A neural network technique to improve computational efficiency of numerical oceanic models”, *Ocean Modelling*, 4, pp. 363-383, 2002.
- [5] V.M. Krasnopolsky, M.S. Fox-Rabinovitz, and D.V. Chalikov, “New Approach to Calculation of Atmospheric Model Physics: Accurate and Fast Neural Network Emulation of Long Wave Radiation in a Climate Model”, *Monthly Weather Review*, v. 133, No. 5, pp. 1370-1383, 2005.
- [6] V.M. Krasnopolsky and M.S. Fox-Rabinovitz, “Complex Hybrid Models Combining Deterministic and Machine Learning Components for Numerical Climate Modeling and Weather Prediction”, *Neural Networks*, 2006, in press
- [7] V. M. Krasnopolsky, et al., “Using Neural Network to Enhance Assimilating Sea Surface Height Data into an Ocean Model”, *Proceedings of the IJCNN2006, July 16-21, 2006, Vancouver, BC, Canada*, 2006, in press
- [8] M.S. Fox-Rabinovitz, V.M. Krasnopolsky, and A. Belochitski, “Neural Network Ensemble Approach for Improving the Accuracy of Climate Simulations that use Neural Network Emulations of Model Physics”, *Proceedings of the IJCNN2006, July 16-21, 2006, Vancouver, BC, Canada*, 2006, in press
- [9] W.W. Hsieh, “Nonlinear principle component analysis by neural networks”, *Tellus*, 53A, pp. 599-615, 2001
- [10] V.N. Vapnik, *The Nature of Statistical Learning Theory*, 189 pp., Springer, New York, 1995.
- [11] F. Chevallier, and J-F Mahfouf, “Evaluation of the Jacobians of Infrared Radiation Models for Variational Data Assimilation”, *J. Appl. Meteor.*, 40, pp. 1445-1461, 2001.
- [12] F. Aires, M. Schmitt, A. Chedin, and N. Scott, “The “Weight Smoothing” Regularization of MLP for Jacobian Stabilization”, *IEEE Trans. Neural Networks*, 10, pp. 1502-1510, 1999.
- [13] F. Aires, C. Prigent, and W.B. Rossow, “Neural network uncertainty assessment using Bayesian statistics with application to remote sensing: 3. Network Jacobians”. *J. Geophys. Res.*, 109, D10305, 2004.
- [14] K. Hornik, M. Stinchcombe, and H. White, “Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks”, *Neural Networks*, 3, pp. 551-560, 1990.
- [15] U. Naftaly, N. Intrator, and D. Horn, “Optimal ensemble averaging of neural networks”, *Network: Comput. Neural Syst.*, 8, pp. 283-294, 1997

- [16] D. Opitz, and R. Maclin, "Popular ensemble methods: an empirical study", *J. Artificial Intelligence Res.*, 11, pp. 169-198, 1999.
- [17] S. Hashem, "Optimal linear combination of neural networks", *Neural Networks*, 10, pp. 599-614, 1997.
- [18] R. Maclin, and J. Shavlik, "Combining the predictions of multiple classifiers: using competitive learning to initialize neural networks", in *Proceedings of the Eleventh International Conference on Artificial Intelligence*, pp.775-780, Detroit, MI, 1995
- [19] A. J. C. Sharkey, "On combining artificial neural nets", *Connection Science*, 8, pp. 299-313, 1996.
- [20] S.V. Barai and Y. Reich, "Ensemble modeling or selecting the best model: Many could be better than one", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 13, pp. 377-386, 1999